

Képfeldolgozást támogató keretrendszer és modulok fejlesztése

Varga Marcell

2013. december 13.

Tartalomjegyzék

1. Feladatkiírás	2
2. Előzetes kutatásról	2
3. Felhasznált eszközök	2
4. Előzetes feladat	4
5. Célkitűzések	9

1. Feladatkiírás

1.1. A feladat címe

Képfeldolgozást támogató keretrendszer és modulok fejlesztése

1.2. A feladat rövid leírása

Könnyen kezelhető vizuális felülettel rendelkező keretrendszer fejlesztése, amely támogatja egyedi képek vizsgálatát és kötegelt feldolgozását. A paraméterezhető képfeldolgozási algoritmusokat modulok biztosítsák a keretrendszer számára.

Kulcsszavak: szoftverarchitektúra, képfeldolgozás, adatszerkezetek, adatkezelés

1.3. Részfeladatok

- Irodalom áttekintése, összefoglalása
- Technológia kiválasztása
- Architektúra tervezése, dokumentálása
- Keretrendszer, implementálása, tesztelése
- Modulok tervezése, implementálása, tesztelése
- A projekthez tartozó dokumentáció és prezentáció elkészítése

1.4. Konzulens

- Lipovits Ágnes, Pannon Egyetem Matematika Tanszék

2. Előzetes kutatásról

Munkát azzal kezdtük, hogy már meglévő közel azonos funkcionalitással rendelkező szoftvereket, szoftvercsomagokat kerestünk. Ez több-kevesebb sikerrel zárult. A legtöbb esetben a képfeldolgozási funkcionalitásokban nem adódott hiány, azonban ezek kalibrálhatósága és a bemenet feldolgozása már problémákat vetett fel.

3. Felhasznált eszközök

Alapvető cél (ahogyan a témakiírásból is kiderül), az hogy egy laboratóriumi felhasználásra alkalmas, dinamikusan bővíthető felxibilis rendszer készüljön el. A fejlesztői környezet és programnyelv választásakor több tényezőt is figyelembe kellett venni.

3.1. Döntési szempontok

A programnyelvnek és fejlesztői környezetnek alkalmasnake kell lennie az alábbi igények több mint elégséges teljesítésére:

- dinamikus moduláris rendszer egyszerűen kialakítható a segítségével
- képfeldolgozási függvénykönyvtárakkal történő ellátottsága jó
- crossplatform
- futásideje elfogadható (biztosítson lehetőséget alacsony és magas szintű párhuzamosításra)
- segítségével egyszerű és ergonomikus GUI készíthető
- korábbi előismeretek

3.2. Fegyvertár

3.2.1. Fejlesztés

- Linux Mint 13 (Maya)
- C++/Qt (5.1.1 GCC 4.6.3 32bit)
- OpenCv 2.4.6.1
- QtCreator 2.8.1
- verziókövetés: Git
- dokumentálás: Latex, Yed

3.2.2. Deploy

Deploy jelenleg első körben ezen a konfigon történik. További iterációk alkalmával Windows-ra (Xp, 7, 8) is készül deploy.

4. Előzetes feladat

4.1. Bag of Words (BoW)

Első fejlesztési iterációban egy Bag of Words kerül megvalósításra. A Bag of Words (BoW) algoritmus használata szövegek, cikkek elemzésében, illetve nyelvészek számára ismert és használt eszköz. Lényege azon az alapvetésen nyugszik, hogy hasonló témájú szövegben hasonló gyakorisággal kerülnek elő megfelelő szavak. Hasonló összefüggés felírható képek esetén is.

4.1.1. Lépései

1. Feature detection: képi tulajdonságok meghatározása - célunk hogy egyedi az adott objektumra, képre jellemző pontokat megtaláljuk
2. Feature extraction: képi tulajdonságok súlyozása - az előző pontban meghatározott tulajdonságok, redundanciájának csökkentése, tehát ne tároljunk és számoljunk felesleges tulajdonság leírókat: pl.: kistávolságban lévő azonos értékű leírók elhagyása, összevonása
3. Szótár felírása: képi leírók összegyűjtése
4. Szótár klaszterezése: leírók csoportosítása

A fenti megoldásból látható, hogy egy bemeneti képen található objektum, objektum csoportok felírhatóak a szótárban található szavakon vett hisztogram segítségével.

4.2. Annotáció

4.2.1. Támogatott annotációs formátumok

A tanításra használt bemeneti nyers képeket annotálnunk kell valamilyen módon. Természetesen nem szükséges mindig saját magunknak összeállítanunk az annotációs adatbázist, így lehetőség van a programban külső adatok átvételére. Jelenleg kettő fő annotációs szintaktika támogatott:

- Pascal VOC: <http://pascallin.ecs.soton.ac.uk/challenges/VOC/>
- Label ME: <http://labelme.csail.mit.edu/Release3.0/>

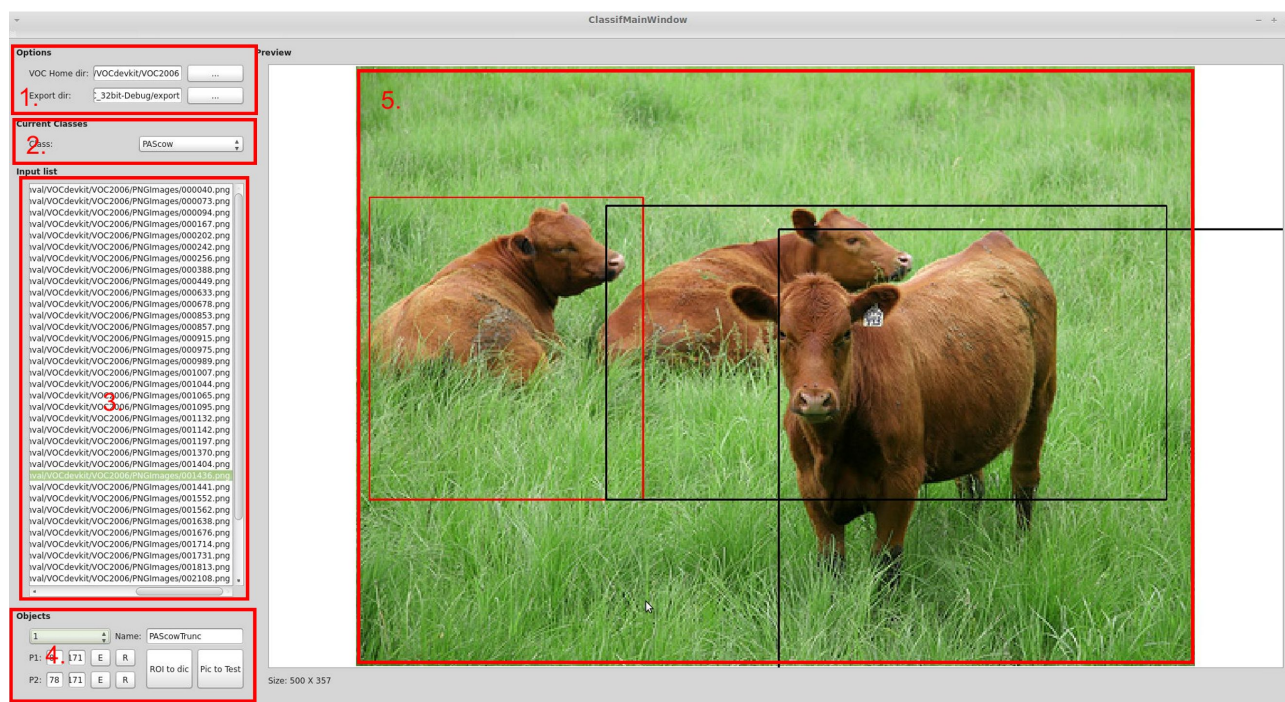
A következőkben táblázatban röviden összehasonlítása következik a két annotációs adatbázis előnyeiről hátrányairól:

	Pascal VOC	Label ME
Formátum:	XML (txt: ver 1.0 esetén)	XML
Annotáció módja:	ROI	polygon
Több osztály:	IGEN	IGEN
Több objektum:	IGEN	IGEN
Validálási info:	NEM	IGEN

Egyértelműen látható, hogy mind a kettő módszerrel a legalapvetőbb műveletek elvégezhetőek, tehát adott osztályú objektum pozíciója meghatározható. A különbség abból adódik, hogy még a Pascal VOC-os adatbázis esetén a bal felső sarok pozíciója tartozik "garantáltan" a megfelelő helyen, addig a Label ME esetén egy polygonnal határozzuk meg az objektum pontos kiterjedését és pozícióját. Megállapítható, hogy a LABEL ME-s annotációk esetén egyértelműbben elkülöníthető az objektum a háttértől már a képi tulajdonságok számításakor is. Viszont sajnos az annotáció elkészítése több időt vesz igénybe.

4.2.2. Annotációs lehetőségek a Classif modulban

Az alábbi képen látható a Classif annotációs formja.



- Kezelő felület:

1. Input/Export könyvtárak és helyek állítása

2. Aktuálisan listázott osztály kiválasztása
3. Aktuális osztályú objektumokat tartalmazó képek listája
4. Aktuális képen található objektumok
5. Aktuális kép



- 4. részletesen:
 1. Aktuális objektum kijelölése (az előző kezelő felületre rajzolt ROI aktív lesz)
 2. Aktuális objektum osztálya - azért szükséges mert egy képen több különböző típusú objektum is lehet
 3. ROI bal felső sarkának koordinátái
 4. ROI jobb alsó sarkának koordinátái
 5. ROI bal felső sarkának pozícionálása (BE/KI - billenő gomb)
 6. ROI jobb alsó sarkának pozícionálása (BE/KI - billenő gomb)
 7. ROI bal felső sarkának pozícionálása (RESET)
 8. ROI jobb alsó sarkának pozícionálása (RESET)
 9. ROIba tartozó pixelek új képként történő küldése szótár kialakításhoz
 10. ROIba tartozó pixelek új képként történő küldése teszt eset kialakításhoz

4.3. Képi tulajdonságok

4.3.1. Detektálás és súlyozás

A képi tulajdonságok meghatározásához az OpenCV több féle lehetőséget biztosít:

- SIFT
- SIFT

- SIFT
- SIFT
- SIFT

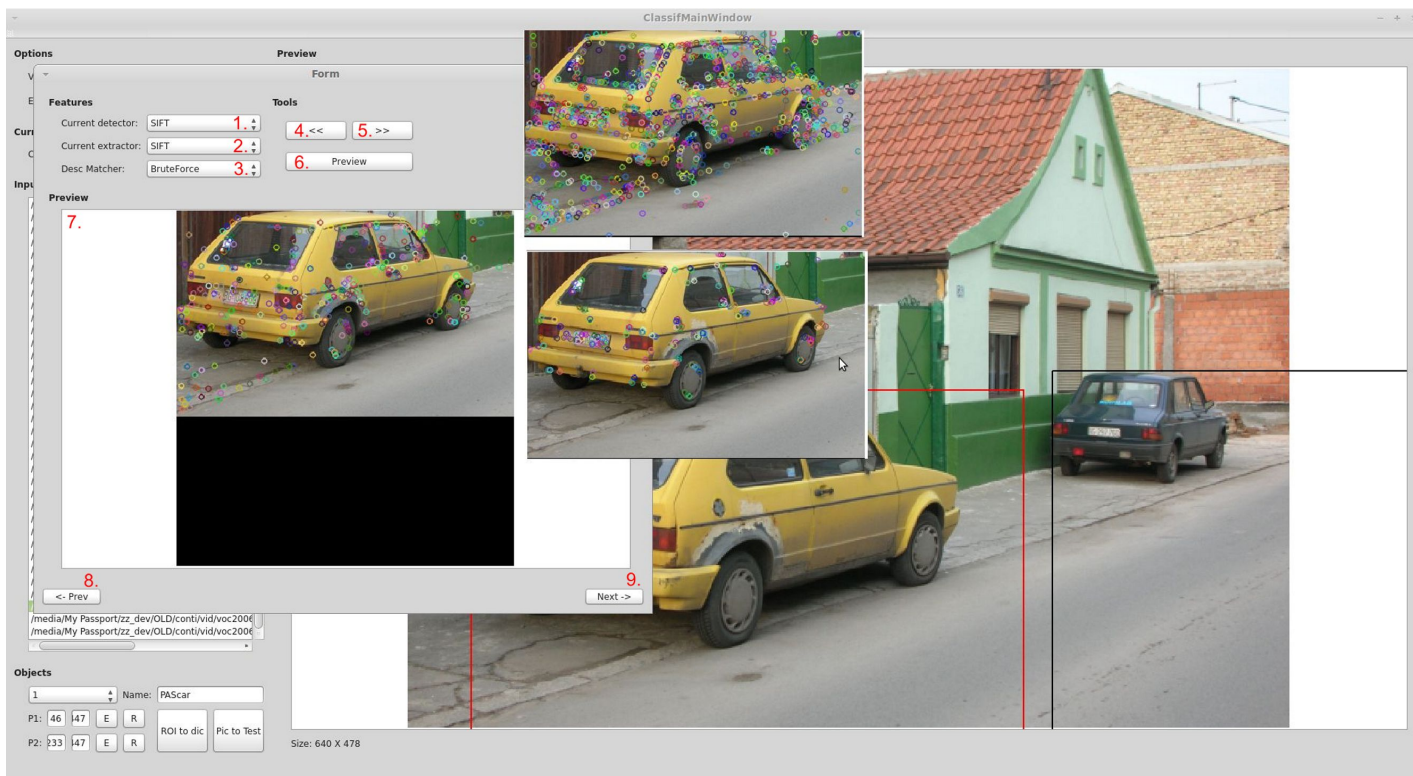
A képi súlyozásához az OpenCV több féle lehetőséget biztosít:

- SIFT
- SIFT
- SIFT
- SIFT
- SIFT

Az lneti kezelő felületen beállítható a kívánt tulajdonság detektáló mód, továbbá továbbá az ezekhez használt súlyozási mód és a minták között történő matchelési módszerek. A kiválasztás után kis előnézeti képen jelenik meg, hogy az adott bemeneti képhez hol milyen leírók kerültek kiszámításra. (A képernyő fotóra a jelenlegi SIFT-SIFT mellé összehasonlítási alapnak felkerült a HARRIS-SIFT és FAST-SIFT párosítás is.) Jelenleg a SIFT-SIFT párosítást használjuk és ez is tesztelt, a jövőben a többi párosítás is tesztelve lesz, illetve további összehasonlítási lehetőségek lesznek a programban.

- Kezelő felület részei:

1. Aktuális feature detector kiválasztása
2. Aktuális feature extractor kiválasztása (SIFT ajánlott)
3. Leírók matchelési módjának meghatározása
4. Előző szótár építő input kérése
5. Következő szótár építő input kérése
6. Beállított detector/extractor/matcher előnézete
7. Előnézeti kép
8. Vissza a szótárépítő inputok áttekintéséhez
9. Előre a szótárépítéshez és gyors teszthez



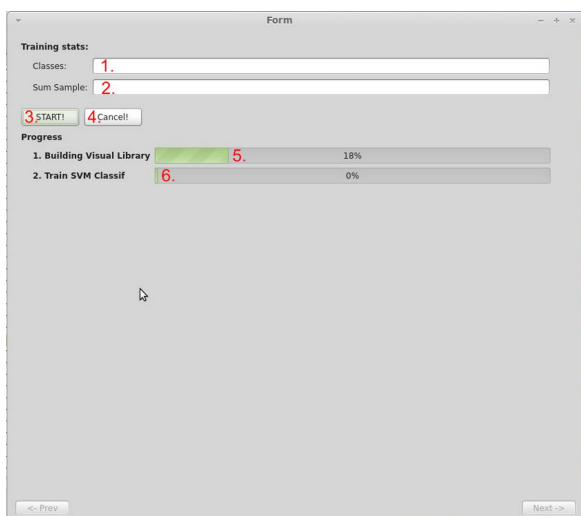
4.4. Szótár építés

4.4.1. Lépései

1. Súlyozott képi információk összegyűjtése
2. Összegyűjtött információk osztályokba szervezése (clustering)

4.4.2. Klaszterezési módszerek

- K-means clustering: <http://shabal.in/visuals/kmeans/1.html>



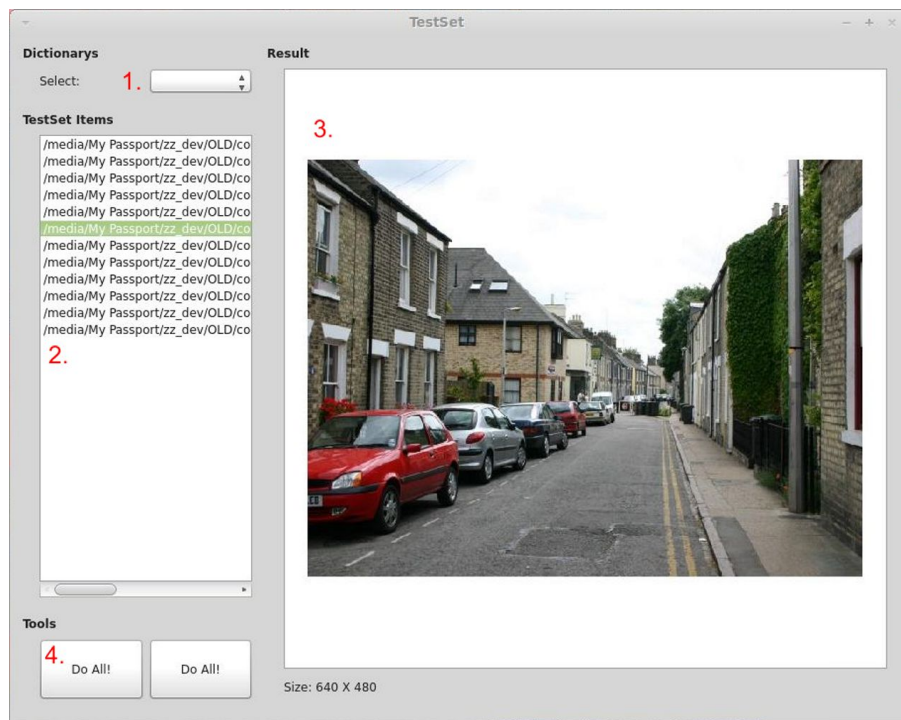
4.4.3. Kezelő felület

1. Aktuálisan a rendszerben lévő kiértékelésre train műveletre sorba állított osztályok száma
2. Összes összegyűjtött minta
3. Start
4. Cancel
5. Aktuális szótár összeállításának folyamata
6. Aktuális inputok vizsgálatának folyamata

Jelenleg az azonosítási folyamat idáig jut el. Tehát maga a szótár elkészül (a bináris mappájába kerül exportálásra, out.txt néven). Azonban a teszt-esetes ellenőrzés már nem fut le.

4.5. Tesztelés

A teszt modul félautomata és automata tesztelésre van előkészítve. A rendszerben lévő szótár fájlok közül kiválasztva a kívántat, továbbá a bemeneti tesztképeket kiválasztva indulhata a teszt. (Természetesen a teszthez nem szükséges előzetesen annotált képeket használni.)



1. Aktuálisan a rendszerben lévő szótárak közül választhatunk célszótárat
2. Aktuális teszt minták

3. Teszt előtt előnézeti képként funkcionál, teszt után vizsgálhatóak az azonosított objektumok
4. Tesztelés indítása
5. Tesztelés megszakítása

5. Célkitűzések

5.1. Bimg

Természetesen mind maga a Classif fejlesztése és tesztelése közben felemrülhetnek bizonyos korábban nem látott igények a főprogrammal kapcsolatban. Elsősorban ezek a funkcionalitások bővítik a listát.

1. Modulok refaktorálásának befejezése
2. Autamatizált tesztesetek készítése (főprogramhoz és pluginhez)
3. Hibás pluginek, paraméterezések kezelése
4. Párhuzamosítás + stressztesztek

5.2. Classif

1. Tesztelési lehetőségek befejezése
2. Lehetőség a belső működésbe történő jobb beavatkozásra: pl.: szótárak kép formában történő exportálása
3. teljes Pascal VOC és Labeled ME támogatás
4. Autamatizált tesztesetek készítése (főprogramhoz és pluginhez)
5. Párhuzamosítás + stressztesztek
6. BoW megközelítésen túl más féle detektálási metódusok támogatása
7. Video + Live stream feldolgozása